



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# VIDEOHERNÍ ARKÁDOVÝ AUTOMAT

## Bakalářská práce

*Studijní program:* B2646 – Informační technologie  
*Studijní obor:* 1802R007 – Informační technologie  
*Autor práce:* **Lukáš Hanuš**  
*Vedoucí práce:* Ing. Martin Rozkovec, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# VIDEOGAME ARCADE MACHINE

## Bachelor thesis

*Study programme:* B2646 – Information Technology  
*Study branch:* 1802R007 – Information Technology  
*Author:* **Lukáš Hanuš**  
*Supervisor:* Ing. Martin Rozkovec, Ph.D.



---

TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky a mezioborových studií  
Akademický rok: 2013/2014

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lukáš Hanuš**  
Osobní číslo: **M10000146**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Videoherní arkádový automat**  
Zadávající katedra: **Ústav informačních technologií a elektroniky**

### **Z á s a d y   p r o   v y p r a c o v á n í :**

1. Seznamte se s historií videoherních automatů a s jejich softwarovými emulátory
2. Navrhněte mechanickou a elektrickou konstrukci videoherního kabinetu včetně ovládacích prvků a mincovníku.
3. Pro elektroniku využijte standardní PC komponenty.
4. Naprogramujte demonstrační hru pro více hráčů využívající prostředků kabinetu

Rozsah grafických prací: Dle potřeby dokumentace

Rozsah pracovní zprávy: cca 30 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

- [1] Charles Petzold, Mistrovství ve WPF, Computer Press, EAN: 9788025121412
- [2] Špaček Jiří, Spielmann Michal: AutoCAD, Computer Press, ISBN: 978-80-251-3120-6
- [3] Build Your Own Arcade Controls wiki, cit. 10. 10. 2013 url: <http://wiki.arcadecontrols.com>

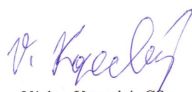
Vedoucí bakalářské práce:

Ing. Martin Rozkovec, Ph.D.

Ústav informačních technologií a elektroniky

Datum zadání bakalářské práce: 12. září 2013

Termín odevzdání bakalářské práce: 16. května 2014

  
prof. Ing. Václav Kopecký, CSc.

děkan



  
prof. Ing. Zdeněk Plíva, Ph.D.

vedoucí ústavu

V Liberci dne 12. září 2013



## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

## **Poděkování**

Tímto děkuji vedoucímu bakalářské práce Ing. Martinu Rozkovcovi Ph.D. za konzultace, cenné rady, podněty a připomínky při zpracování mé bakalářské práce.

Dále bych chtěl poděkovat panu Ing. Miroslavu Vavrouškovi za konzultace ohledně programování v jazyce C#.

Velké díky patří rovněž mému kamarádovi Tomášovi Rejdákovi, který se zabývá truhlářinou, za jeho pomoc při konstrukci herního kabinetu a za poskytnutí profesionálního nářadí.

Také děkuji svému otci, který mi poskytl svůj prostor v dílně a byl mi nápomocen.

## **Abstrakt**

Tato práce se zabývá problematikou návrhu a výroby videoherního arkádového automatu, který by měl splňovat bezchybnou funkčnost a co možná nejvyšší hratelnost.

V úvodu práce se zabývám historií videoherních automatů, softwarových emulátorů, výběrem nejvhodnější videohry a možností použití různých programovacích jazyků. Hlavním obsahem celé práce je konstrukce a prezentace videoherního automatu. V první části je popsána truhlářská výroba kabinetu. Dále se zabývám výběrem vhodných ovládacích komponentů, které spolu musí bezproblémově komunikovat, aby byla zajištěna elektronická funkčnost. V závěru mé práce popisuji naprogramování arkádové hry. Věřím, že zaujme a přiláká mnoho lidí, kteří se chtějí ve svém volném čase pobavit.

## **Klíčová slova**

herní automat, arkáda, emulátor, mincovník, Csharp

## **Abstract**

The Bachelor's thesis deals with the project and production of videogame arcade machine which should fulfill the functionality without errors and the highest possibility of playability.

At the beginning of the thesis, I describe history of game machines, emulation software, the selection of the most suitable video game and the possibility of using different programming languages. The main subjects of this thesis are construction and presentation of the video game machine. The first part describes the carpentry production of a box. Further, I deal with the selection of the suitable control components which have to operate without any problem in order to ensure the electronic functionality. In conclusion, I describe the programming of the arcade game. I believe that this machine will interest and attract wide range of population who would like to entertain themselves in their free time.

## **Key words**

slot machine, arcade, emulator, coin acceptor, Csharp

# Obsah

1 Úvod.....	9
2 Rešerše.....	10
2.1 Arkádová hra.....	10
2.2 Výběr arkádové hry.....	10
2.3 Výběr vhodného programovacího jazyka a vývojového prostředí.....	11
2.4 Stručná Historie arkádových automatů.....	11
2.5 Přehled emulátorů.....	13
3 Řešení.....	14
3.1 Konstrukce kabinetu.....	14
3.2 Mechanická a elektrická realizace.....	15
3.3 Tlačítka.....	16
3.4 Podsvícení tlačítek.....	19
3.5 Mincovník.....	21
3.5.1 Programování mincovníku.....	23
3.6 Klávesnicový emulátor.....	25
3.7 Výběr komponentů a PC sestava.....	27
3.8 Softwarové vybavení .....	28
3.9 Tvorba aplikace.....	29
4 Závěr.....	32
5 Seznam použité literatury a dalších zdrojů.....	34
6 Přílohy.....	35

## Seznam tabulek

Tabulka 1: Technické parametry mincovníku CH-926.....	22
Tabulka 2: Konfigurační tabulka klávesnicového emulátoru.....	27
Tabulka 3: Přehled vybraných komponentů.....	27
Tabulka 4: Finanční náklady herního automatu.....	33

## Seznam ilustrací

Ilustrace 1: Blokové schéma ovládacích prvků automatu.....	16
Ilustrace 2: Technický výkres tlačítka.....	17
Ilustrace 3: Tlačítko.....	17
Ilustrace 4: Rozvržení tlačítek.....	18
Ilustrace 5: Schéma zapojení tlačítek.....	18
Ilustrace 6: Voltampérová charakteristika led-diody.....	19
Ilustrace 7: Schéma zapojení led-diod.....	20
Ilustrace 8: Technický výkres mincovníku CH-926.....	22
Ilustrace 9: Detailní ilustrace mincovníku CH-926.....	22
Ilustrace 10: Popis mincovníku CH-926.....	23
Ilustrace 11: Klávesnicový emulátor ARCADE I-PAC2 FS32.....	26

## Seznam příloh

Příloha A: Technický výkres kabinetu – pohled shora.....	35
Příloha B: Reálná ilustrace kabinetu - pohled shora.....	36
Příloha C: Technický výkres kabinetu - pohled zepředu.....	37
Příloha D: Reálná ilustrace kabinetu - pohled zepředu.....	38
Příloha E: Technický výkres kabinetu - boční pohled.....	39
Příloha F: Reálná ilustrace kabinetu - boční pohled.....	40
Příloha G: Ukotvení monitoru a počítače.....	41
Příloha H: Vystužení kabinetu.....	42
Příloha I: Konfigurační soubor hry.....	43

# 1 Úvod

Toto téma jsem si zvolil proto, že jsem s počítačovými hrami přišel do styku už jako dítě a neustále se jimi zabývám. Největší mojí motivací byl již fungující nevýherní automat v jednom klubu v areálu Harcov v Liberci, který mě ihned uchvátil a už jsem přemýšlel o jeho možné inovaci. Také si myslím, že je škoda, že éra těchto automatů už skončila a dostávají se do popředí výherní automaty, které mají špatný sociálně-ekonomický vliv. Na základě výsledků této práce a vyrobením prvního prototypu bych rád uskutečnil malosériovou výrobu těchto videoautomatů.

Cílem této práce je navržení mechanické a elektrické konstrukce videoherního kabinetu, včetně ovládacích prvků a mincovníku, aby byl co nejjednodušší a nejlevnější na výrobu. Hardwarový důraz se klade na PC komponenty. K řešení softwarové stránky nebyl vysloveně zadán žádný operační systém, ani samotná hra, tudíž je nejdříve nutné vybrat nějakou populární arkádu, která se postará o maximální hratelnost, aby byla zajištěna nejvyšší přízeň ze strany samotných uživatelů. Dále je potřeba zvolit vhodný programovací jazyk pro realizaci samotné arkádové hry.

## 2 Rešerše

V této kapitole stručně popisuji co je to arkádová hra. Dále se zabírám výběrem mé arkádové hry, vhodným použitím programovacího jazyka a vývojovým prostředím. V poslední části této kapitoly se zabývám historií videoherních automatů a softwarovými emulátory.

### 2.1 Arkádová hra

Arkáda je určitý žánr počítačové hry. Je založený na jednoduchém a nápaditém konceptu.

Arkády se nejčastěji hrají na kola, přičemž každé kolo má vyšší obtížnost. Některé hry bývají omezeny časem. Právě tento druh her byl dříve velice oblíbený na videoherních automatech. S postupem času se začal rovněž objevovat na osobních počítačích. U tohoto druhu her je klíčová hbitost pohybu prstů po klávesnici. U hráčů rozvíjí logické myšlení, předvídání, a i trpělivost.

V arkádách záleží také na šikovnosti a důvtipu. Velké množství vydaných her na stejném principu mělo za následek rozdělení těchto her do různých kategorií podle společných znaků. Mezi nejrozšířenější kategorie her patří například hry bojové, plošinové, sportovní, závodní a logické. V arkádách se stala také velmi populární hra pro dva a více hráčů na jednom počítači, a také hra v módu split screen až pro čtyři hráče.

### 2.2 Výběr arkádové hry

Po uvážení všech možných variant a ohlasů ze strany konečných zájemců o samotnou hru jsem dospěl k závěru, že by to měla být hra určená pro co největší počet hráčů, aby našla zastání na klasických veřejných místech, zejména barech, kde se chtějí lidé bavit a přijít na jiné myšlenky. Po vyloučení všech her pro jednoho hráče jsem měl prvního kandidáta a v podstatě jasno. Bomberman, původně známý jako Dynablaster od vývojářů firmy Hudson Soft [1], byl určitě jednou z nejpopulárnějších arkád

nejenom mého dětství, ale také hrou, na kterou si vzpomene snad každý jedinec mé generace. Další alternativou se nabízelo „Liero“ [2], „Tunneler“, či novodobější „Bulánci“ [3]. Ale to až případně v budoucí rozšířené verzi automatu.

## **2.3 Výběr vhodného programovacího jazyka a vývojového prostředí**

Pro mnou zvolenou demonstrační hru bylo zapotřebí objektově orientovaného jazyka, který umožní herní algoritmus naprogramovat. V dnešní době máme k dispozici celou řadu těchto jazyků např. Java, C++, C#, Visual Basic, Simula. Zde jsem vybíral na základě mých dosavadních zkušeností. Nejpřirozenějším a nejpohodlnějším jazykem se pro mě jevil C#, který je vysokoúrovňový objektově orientovaný jazyk, vyvinutý firmou Microsoft zároveň s platformou .NET Framework. Tento jazyk vychází z jazyku C++ a Java.

Nyní jsem potřeboval vývojové prostředí, které mi poskytne pohodlné psaní syntaxe v jazyce C#. Všechny mé dosavadní projekty jsem psal ve vývojovém prostředí Microsoft Visual Studio. Protože jsem potřeboval prostředí zaměřené na vývoj her, rozhodl jsem se použít platformu Microsoft XNA. Tato platforma v podstatě spojuje již zmíněné Visual studio s XNA Framework, který obsahuje knihovny třídy, určené pro vytváření 2D a 3D her.

## **2.4 Stručná Historie arkádových automatů**

První mechanické herní automaty se začaly objevovat už v šedesátých letech devatenáctého století. Například to byl simulátor ponorka Periscope od firmy Sega z roku 1966, nebo závodní Grand Prix z roku 1969. Arkádové automaty se rozšířily na západě v sedmdesátých letech a postupně začaly zaplavovat veřejná místa jako bary, kina, diskotéky a pizzerie. Nakonec byla zhotovena specializovaná zábavná střediska jako Sega World v Londýně nebo GameWorks ve Spojených státech amerických. Velký průlom však přišel s nástupem hry Pong, kterou uvedla na trh společnost Atari v roce 1972. Jako první velký hit je označována hra Space Invaders z roku 1978. Od tohoto roku zažívaly arkádové hry doslova raketový vzestup. V dnešní době si už nikdo z nás nedovede představit vhadzovat mince do veřejných automatů, když si může libovolnou



hru zahrát doma na svém osobním počítači. Ale tehdy to možné nebylo. S první domácí herní konzolí, označovanou jako Atari 2600 přišla firma Atari na přelomu sedmdesátých a osmdesátých let. Ovšem tato konzole s neuvěřitelným rozlišením 192x160 bodů a čtyřmi barvami byla oproti tzv. Arkádovému Donkey Kong, který měl rozlišení 224x256 a 256 barev doslova dětská hračka. Arkádové automaty nijak neohrozil ani nástup domácích počítačů jako ZX Spectrum a C64 nebo herní konzole NES od Nintendo v osmdesátých letech. Automaty byly vždy o krok napřed. Domácí osmibitové počítače byly poháněny procesory s taktovací frekvencí jedna až čtyři Mhz. U videoherních arkádových automatů to byla však Motorola 68010 o výkonu sedmi až deseti Mhz. Tento typ procesorů se začal objevovat o několik let později v domácích strojích Atari ST. Ale ani s tímto průlomem nedokázaly automaty konkurovat v rozlišení, velikosti displeje a množství barev. Dalším důležitým aspektem zde byla cena. Po uvedení na trh Amiga byla jeho cena neuvěřitelných 600 dolarů. Za tuto závratnou cenu si mohli hráči zahrát a užít stejnou hru v daleko lepší kvalitě až 2400x v některé herně na arkádovém automatu. Aby si výrobci arkádových automatů udrželi pozici na trhu, soustředili se na něco, co si samotný hráč doma dovolit nemůže. Vybavovali automaty ovladači a simulátory, které zacházely téměř do virtuální reality. Již první tyto stroje připomínaly díky své monstrózní velikosti kokpity aut nebo letadel, které lidé ve hře ovládali. Postupem času tyto kabinety výrobci ještě vylepšili o hydrauliku, která v průběhu hry všemožně lomcovala s celým zařízením.

V devadesátých letech arkádovým hernám zlomil vaz nástup 3D grafických karet. I když to byly právě arkádové automaty, které přinášely jedinou možnost si zahrát tehdejší 3D hry. Tehdy vyvinula neznámá firma 3dfx čip pro akceleraci 3D grafiky Voodoo. Tyto čipy se zpočátku využívaly právě v automatech, ale později se rozšířily i do osobních počítačů. 3D hry existovaly už před nástupem Voodoo, ale tyto akcelerátory dokázaly zvýšit celkový výkon a vyhladit pixelové textury. Odezva vývoje herních automatů byla obrovská a přišla s opravdu realistickými simulátory. Bohužel příchod nové generace konzolí znamenal konec pro arkádový průmysl. To ovšem neznamená, že se v dnešní době nemůžeme setkat ještě s těmito arkádami. Nedaleko Prahy se nachází muzeum arkádových her, kde má každý návštěvník možnost zahrát si nepřeberné množství hitů z tehdejších her. [4]

## 2.5 Přehled emulátorů

Emulátor je software, který umožňuje spouštění programů a her v jiných operačních systémech nebo platformách, než pro které byly původně vytvořeny .

Nejrozšířenějším domácím počítačem osmdesátých let byl Sinclair ZX Spectrum. Tento oblíbený osmibitový počítač měl procesor Z80 a 48kB operační paměti. Nejlepším emulátorem Sinclair ZX Spektrum je označovaný ZXAM 2.0b, který umožňuje spuštění nepřeberného množství her určených pro tento stroj. [5]

Jedním z nejrozšířenějších emulátorů je MAME (Multiple Arcade Machine Emulator), který emuluje herní automaty všech možných druhů. MAME umožňuje spuštění aplikací na dnešních osobních počítačích. Aplikace jsou zpravidla zabaleny do jednoho souboru, který se označuje anglickým názvem ROM. Tento emulátor dokáže momentálně emulovat přes 2600 unikátních a přes 4600 klasických arkádových her, které pocházely z let sedmdesátých, osmdesátých a devadesátých. MAME si klade za cíl nenechat tyto legendární hry zapadnout v historii. [6]

Z novodobějších emulátorů stojí za zmínku velice oblíbený DOSBox, který emuluje operační systém MS-DOS, pro který bylo vytvořeno také obrovské množství her.

## 3 Řešení

Tato kapitola obsahuje kompletní tvorbu kabinetu od jeho fyzické konstrukce, přes mechanickou obsluhu až po naprogramování demonstrační hry.

### 3.1 Konstrukce kabinetu

Prvním krokem celé práce bylo vhodné navržení a konstrukce kabinetu [7]. Zde bylo zapotřebí rozumně zvolit rozměry a tvar, aby se dal automat ovládat čtyřmi lidmi zároveň. Nechal jsem se inspirovat už existujícím nevýherním automatem z jednoho libereckého klubu. Proto mé myšlenky směřovaly k přenosné bedně, u které budou stát čtyři hráči, na každé straně jeden. Každý bude mít k dispozici ovládací tlačítka a bude schopen sledovat tentýž monitor za stejných podmínek. Z důvodu jednoduchosti na výrobu jsem zvolil čtvercový půdorys (viz příloha A).

Dalším důležitým aspektem byla volba materiálu, ze kterého bude kabinet zhotoven. Kvůli nárokům na bytelnost jsem se rozhodl pro dřevo, přesněji řečeno pro překližku o tloušťce 20mm, což je kompozitní deska, která je vyrobena ze tří nebo více vrstev loupaných nebo krájených dřív, které jsou na sebe lepeny křížem. Jedná se tedy o velmi masivní dřevo, které je odolné vůči nežádoucím vlivům, zejména ze strany koncových uživatelů.

Díky známosti z řad truhlářů, jsem měl možnost výhodné koupě materiálu, zapůjčení profesionálního nářadí a zhotovení desek na míru. Za pomoci vrutů jsem upevnil desky k sobě a vznikla tak kostra kabinetu. Hlavní spoje jsem opatřil speciálním lepidlem na dřevo, vyztužil je dřevěnými trámky (viz příloha H) a pro lepší pevnost jsem použil ještě lamely. V další fázi jsem pomocí přímočaré pily udělal otvory pro madla, monitor a mincovník a vykrúžovacím vrtákem jsem vykroužil otvory pro tlačítka. Pro lepší manipulaci a snadnější přesun kabinetu jsem použil kovová madla (viz příloha F) pro reproboxy zapuštěná dovnitř. Pro snadnější přístup do vnitřku kabinetu bylo nutné ho opatřit dvířky připevněnými na pantech a zamykatelných petlicí (viz příloha D), která dodává celému automatu retro vzhled.

Celou kostru kabinetu bylo třeba pečlivě a důkladně obrousit do hladka. Pomocí

frézky jsem zaoblil všechny rohy a hrany. Následně jsem celý kabinet třikrát natřel latexovou barvou, která sloužila jako penetrace před finálním nátěrem černou tabulovou barvou. Na závěr jsem celou bednu přestříkal matným lakem a otvor pro monitor jsem překryl čtvercovou tabulkou z plexiskla.

V další fázi jsem kabinet osadil ovládacími a zobrazovacími prvky. Tlačítka jsou do vykroužených otvorů upevněna pomocí závitů a plastových matic. K upevnění madel jsem použil klasické vruty a monitor jsem do kabinetu připevnil pomocí kurty s ráčnou (viz příloha G). Po obvodu celého monitoru jsem ještě umístil trámky (viz příloha H), které zamezují pohybu monitoru. Mincovník jsem do vyříznutého otvoru připevnil originálními šrouby, které byly součástí balení (viz příloha F). Nakonec přišel na řadu samotný počítač, který jsem taktéž zajistil kurtou s ráčnou (viz příloha G).

### **3.2 Mechanická a elektrická realizace**

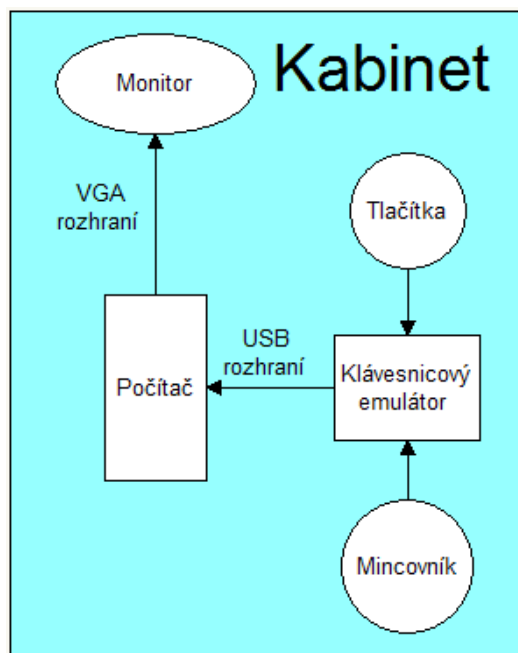
V této fázi jsem měl hotové prakticky všechny truhlářské práce a připravený prázdný kabinet. Nyní bylo třeba automat osadit sobě kompatibilními komponenty, které se postarají o mechanickou a elektronickou funkčnost celého zařízení.

Jedná se zde o videoherní automat, na kterém se bude hrát hra určená pro dva až čtyři hráče, proto je nezbytné použít nějaké ovládací zařízení pro každého abonenta.

Samoobslužná zařízení, jako např. automat na kafe, pití, bagety, stolní fotbálek apod., se neobejdou bez zařízení, které přijímá mince. Pro tento účel byl sestrojen mincovník, který funguje na principu akceptování určitého peněžního obnosu za poskytnuté služby. Proto jsem i já kabinet opatřil tímto zařízením.

Aby tlačítka a mincovník komunikovaly se samotným softwarem, bylo třeba převést výstupní analogový signál z těchto zařízení do digitálního. K řešení tohoto problému jsem se rozhodl využít tzv. USB klávesnicového emulátoru, který emuluje stisk kláves a převádí jej do digitální podoby a zasílá do počítače pomocí USB portu. Nyní byl zapotřebí počítač, který tento digitální signál spolu s hrou zpracuje. Posledním prvkem kabinetu je zobrazovací zařízení v podobě monitoru.

Hierarchie všech zmíněných prvků je zobrazena na následujícím blokovém schématu.



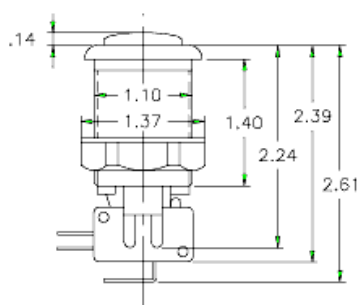
*Ilustrace 1: Blokové schéma ovládacích prvků automatu*

K realizaci většiny zapojení jsem použil UTP kabel kategorie 5e, který je pro dané obvody zcela dostačující, ale hlavně běžně dostupný. Tento druh vodiče se používá v ethernetových rozvodech, obsahuje čtyři kroucené páry (červený, zelený, modrý a hnědý). Veškeré spoje jsou pečlivě proletovány, aby někde nedocházelo k tzv. studeným spojům.

### 3.3 Tlačítka

Nedílnou součástí kabinetu jsou ovládací rozhraní pro jednotlivé hráče. K realizaci mé hry jsem potřeboval pro každého hráče zajistit čtyři tlačítka pro pohyb do čtyř směrů a jedno tlačítko na libovolnou funkci. Na tomto místě se nabízely dvě varianty provedení. Držet se retro stylu a zvolit populární pákové ovladače (joystick) s jedním tlačítkem, které by umožňovaly pohyb do všech čtyř stran a jednu funkci. Opět jsem se zamyslel nad chováním hráčů k automatu. Kdybych se přiklonil k této

verzi ovládání, musel bych volit opravdu velice bytelný joystick, který by byl imunní vůči většině nepříznivých vlivů ze strany koncových uživatelů. Tomu by potom odpovídala pochopitelně i cena. Po vyloučení pákového ovladače jsem se rozhodl použít klasická kulatá tlačítka. Myšlenka byla celkem jednoduchá a jasná. Začal jsem pokukovat po internetu a obcházet kamenné obchody s elektronikou. Byl jsem značně vyveden z míry, protože sehnat přijatelná tlačítka, není ani v dnešní době vůbec jednoduché. Dokonce mi bylo v několika obchodech řečeno, že tlačítka s těmito parametry se vůbec nevyrábí. V závěru usilovného hledání jsem konečně našel kandidáta, který se přibližoval k mé požadované představě, ale po zjištění cenových nároků na jedno tlačítko, jsem byl nucen zavítat na mezinárodní internetové obchody. K mému překvapení jsem byl mile potěšen, neboť arkádovými tlačítky se to zde doslova hemžilo. Cena byla taktéž velmi přijatelná. Okamžitě jsem vytvořil objednávku a vyčkával na doručení. Zásilku jsem měl celkem rychle na stole. Nyní mi už nic nebránilo v zabudování tlačítek do vrchní desky kabinetu, kde jsem se opět musel malinko pozastavit.



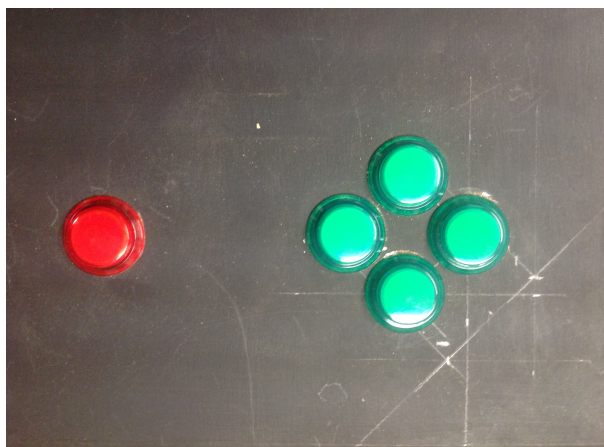
*Ilustrace 2: Technický výkres tlačítka*



*Ilustrace 3: Tlačítko*

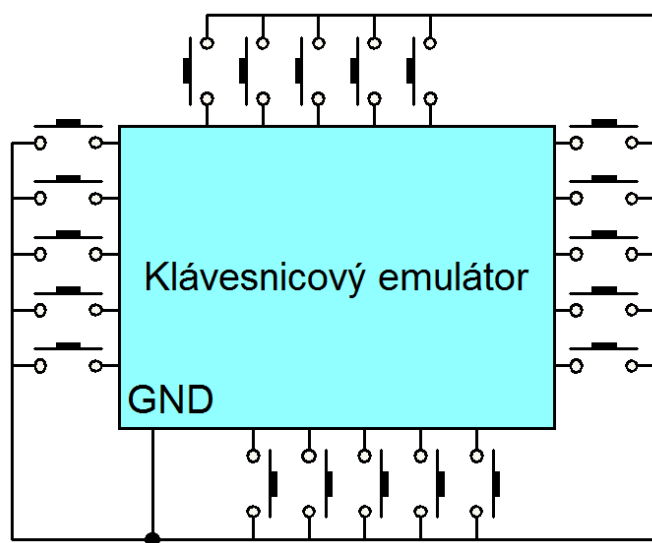
Důležitým aspektem zde bylo vhodné rozvržení a umístění tlačítek tak, aby se hra co nejpohodlněji a nejefektivněji ovládala. Moje první myšlenka směřovala ke standardnímu uspořádání šipek jako na počítačové klávesnici. Ale vzhledem k větším rozměrům tlačítek by se mohlo stát, že někteří jedinci, kteří mají menší rozpětí prstů, zejména děvčata, by shledali ovládání nepohodlným. Nakonec jsem se rozhodl pro použití kosočvercového rozložení ovládacích tlačítek, které bylo rozhodně efektivnější. Tlačítka pro pohyb do stran byla vyřešena, zbývalo tedy pouze vybrat

vhodné místo pro funkční tlačítko. Vzhledem k tomu, že většina hráčů je z klasických arkádových her zvyklá na ovládací tlačítka vpravo a funkční tlačítko vlevo, zvolil jsem i já tento způsob rozvržení.



*Ilustrace 4: Rozvržení tlačítek*

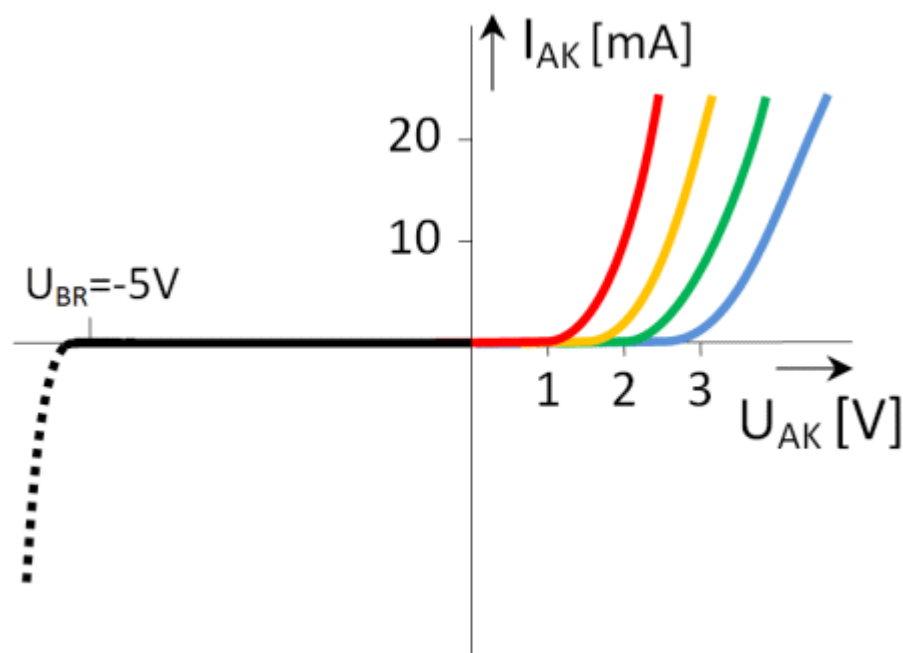
Následující schéma znázorňuje zapojení všech tlačítek.



*Ilustrace 5: Schéma zapojení tlačítek*

### 3.4 Podsvícení tlačítek

Pro upoutání větší pozornosti na herní kabinet jsem opatřil všechna tlačítka podsvícením. Toto podsvícení mají za úkol standardní LED – diody. Využívám pět různých barev tlačítek. Modrou, zelenou, bílou a žlutou pro pohyb do čtyř směrů a pro funkční tlačítko červenou. Proto i barva diod musela být shodná. Problém nastal při výběru vhodného napájení a stylu zapojení. Kabinet jsem se snažil nezaplňovat zbytečnými komponenty, takže jsem hledal zdroj napájení u již použitých součástí. Nejjednodušším řešením se jevilo využití zdroje energie přímo z počítače. Většina hardwaru v počítači pracuje s úrovní napětí 5V a 12V. Tyto dvě úrovně nalezneme v tzv. Molex konektoru. Nyní bylo třeba vhodně zvolit velikost napětí. Podle následující voltampérové charakteristiky led – diody je patrné, že každá barva využívá jinou úroveň napětí, řekněme od 1,5V – 3,5V.



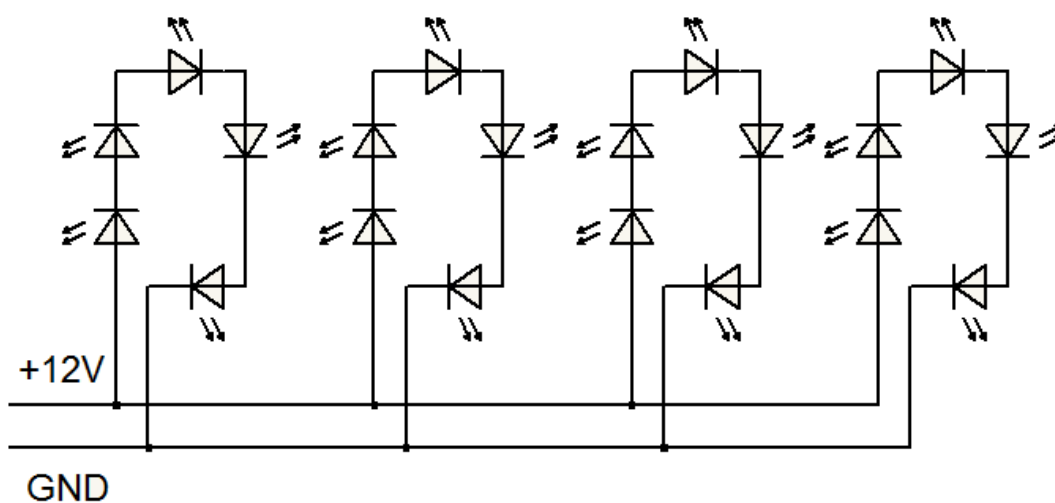
*Ilustrace 6: Voltampérová charakteristika led-diody*



Tyto dvě hodnoty jsem zprůměroval a dostal jsem 2,5V. Takové napětí jsem k dispozici neměl, proto bylo potřeba správně navrhnout zapojení, při kterém by se na každou z dvaceti jednotlivých diod dostalo požadovaného napětí 2,5V. Na tomto místě jsem uplatnil Ohmův zákon, který vyjadřuje vztah mezi elektrickým odporem, napětím a proudem. Rozdělení napětí na led-diodách je přímo úměrné proudu, který jimi protéká. Zde jsem aplikoval dva jednoduché vzorce pro řazení rezistorů.

$$R_p = \frac{R_1 R_2 \dots R_n}{R_1 + R_2 + \dots + R_n} \qquad R_s = R_1 + R_2 + \dots + R_n$$

Výsledkem bylo paralelní zapojení čtyř okruhů, které se skládaly z pěti sériově zapojených diod, jak je znázorněno na schématu níže.



*Ilustrace 7: Schéma zapojení led-diod*

### 3.5 Mincovník

Mincovník je elektronické zařízení, které slouží ke zjištění hodnoty a pravosti mincí. U starších zařízení se pravost ověřovala metodou zjištění váhy a rozměrů, ale ukázala se jako velice neefektivní, protože se začala obcházet a zneužívat. Tyto slabiny se pochopitelně nelíbily majitelům a nájemcům automatů, ve kterých byly tyto mincovníky použity. Proto se začaly využívat jiné bezpečnější fyzikální metody. Jednou z těchto metod je metoda založená na principu magnetismu, kdy magnetické pole je v okolí každého druhu mince rozdílné.

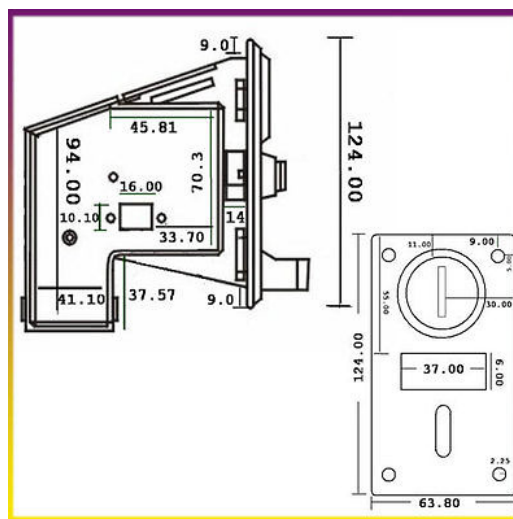
Pochopitelně i já jsem se snažil vyhledat co nejspolehlivější, ale zároveň cenově dostupný mincovník, který by se nedal jednoduše obelstít. Začal jsem na internetu, kde jsem navštívil několik obchodů, ale vše vedlo ke stejnému problému. Cena byla opravdu hodně vysoká. Poptával jsem se proto u svých přátel, kteří pracují alespoň ve vzdáleně podobném oboru. Zjistil jsem, že jedna nejmenovaná firma obdobné mincovníky likviduje. Netrvalo dlouho a několik jsem jich měl doma. Ale bohužel dokumentace k nim nebyla k dohledání. Po několika neúspěšných pokusech uvedení mincovníku do provozu jsem tuto variantu zavrhl.

Nezbývalo mi nic jiného než opět navštívit zahraniční internetové obchody, kde jsem ihned narazil na vhodné kandidáty za minimální cenu. Pro jednoduchost jsem se přiklonil k mincovníkům, které nevracejí peníze.

Výhercem se stal mincovník označený jako CH-926, který se používá v nejrůznějších zařízeních ve Spojených státech amerických. Toto zařízení je schopné přijímat současně až šest různých druhů mincí z celého světa. Typ CH-926 je založený na třech rozpoznávacích algoritmech - váha, velikost a hlavně materiál mince, který je ověřovaný indukčními cívkami. Výstupem mincovníku je pulzní signál, který přesně splňuje moje požadavky. Vstupní napětí je 12V stejnosměrného signálu. Díky této napěťové hladině jsem mohl opět využít jako zdroj elektrické energie Molex konektor.

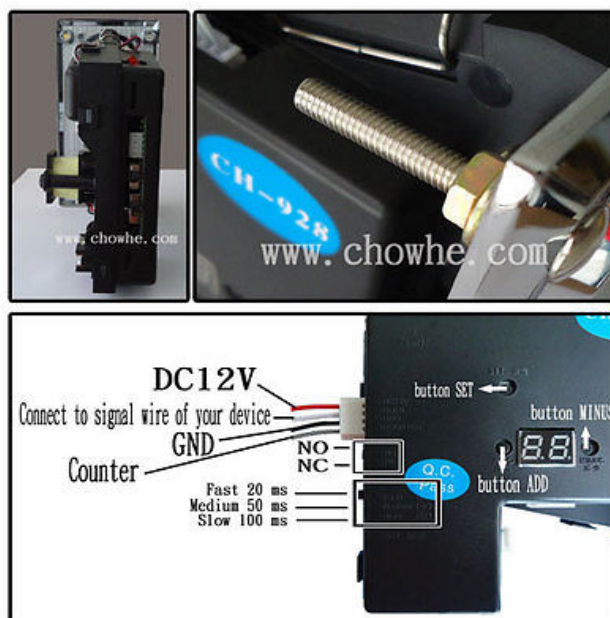
Tabulka 1: Technické parametry  
mincovníku CH-926

Průměr mince	15mm~32mm
Tloušťka mince	1,2mm~3,8mm
Atmosférický tlak	86Kpa - 106Kpa
Přenost identifikace	99,5%
Pracovní napětí	DC +12V $\pm$ 10%
Pracovní proud	65mA $\pm$ 5%
Pracovní vlhkost	$\leq$ 95%
Výstupní signál	pulzní
Rychlost	$\leq$ 0,6s

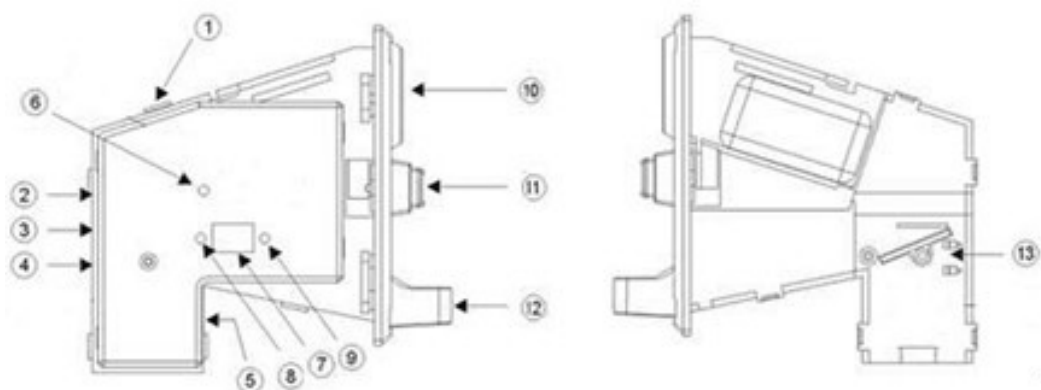


Ilustrace 8: Technický výkres  
mincovníku CH-926

Zařízení disponuje integrovaným inteligentním procesorem, který se stará o naprosto jednoduché naprogramování bez použití dalších zařízení jako je počítač.



Ilustrace 9: Detailní ilustrace mincovníku CH-926



*Ilustrace 10: Popis mincovníku CH-926*

1. LED dioda – světelná indikace
2. čtyřpinový vstupně/výstupní konektor
3. dvoupolohový přepínač „NO“, nebo „NC“
4. třípolohový přepínač pro rychlost výstupního signálu – rychlý: 20ms, střední: 50ms a pomalý: 100ms
5. dvoupinový přídavný vstup
6. tlačítko „SETUP“
7. LED display
8. tlačítko „ADD“ (plus)
9. tlačítko „MINUS“ (mínus)
10. vstupní slot na mince
11. tlačítko pro vyžádání mince zpět, případně při zaseknutí mince
12. výstupní slot pro mince
13. pozice elektromagnetického blikání

### **3.5.1 Programování mincovníku**

Základní naprogramování mincovníku probíhá ve dvou hlavních krocích. V případě potřeby úpravy výstupního pulzu se dá využít ještě třetí krok.

## 1. Nastavení parametrů mincí

V prvním kroku se definuje počet různých mincí, který bude mincovník schopen přijímat, počet mincí na vzorkování, počet výstupních pulzů každé mince a přesnost algoritmů pro vyhodnocování.

**a)** Prvním krokem bylo nastavení počtu druhů mincí, které bude mincovník schopen přijímat, tj. Pět (1 Kč, 2 Kč, 5 Kč, 10 Kč, 20 Kč).

**b)** V tomto bodu jsem zvolil počet vzorků (max. třicet) dané mince pro její správné rozlišení, které se dále používá v druhém kroku. Pro vyšší přesnost jsem použil u každého druhu mince dvacet vzorků.

**c)** Dále jsem nastavil kolik výstupních pulzů (max. padesát) bude pro danou minci. Pro minci v hodnotě 1 Kč jsem zvolil jeden pulz, pro minci v hodnotě 2 Kč dva pulzy, pro minci v hodnotě 5 Kč pět pulzů, pro minci v hodnotě 10 Kč deset pulzů a pro minci v hodnotě 20 Kč výstupních pulzů dvacet.

**d)** V posledním kroku jsem navolil přesnost daných algoritmů pro rozpoznání dané mince (jedna až třicet, kde jedna je nejpřesnější). Pro optimální přesnost jsem každé minci nastavil hodnotu osm.

**e)** První minci jsem měl nastavenou, zbývalo donastavit ostatní čtyři mince opakováním bodů **b)** až **d)**.

**f)** Po nastavení všech mincí stačilo pro uložení celý mincovník restartovat (odpojením a následným připojením k napájecímu zdroji).

## 2. Vzorkování

V druhém kroku jsem musel všech pět mincí navzorkovat, aby si mincovník uložil o jednotlivých mincích nezbytné informace.

**a)** Po dobu několika sekund jsem stiskl tlačítko „SET“ a na zobrazovacím zařízení se objevilo „A1“ (první mince – 1 Kč).

**b)** Nyní jsem začal do mincovníku postupně vhazovat mince. Když jsem

vhodil všech dvacet vzorků první mince, objevilo se na zobrazovacím zařízení „A1“. Tím se ukončilo vzorkování první mince.

c) Opětovným stlačením tlačítka „SET“ po několik sekund se na zobrazovacím panelu objevilo „A2“ (druhá mince – 2 Kč).

d) Dále jsem pokračoval bodem b) dokud jsem nenavzorkoval všechny mince.

### 3. AP mode

Tento třetí krok je zcela volitelný a slouží k úpravě požadovaného výstupního pulzu. Jelikož jsem pro spuštění hry zvolil částku 20 Kč, bylo nutné mincovník nastavit tak, aby zasílal jeden výstupní pulz, při dosažení součtu dvacet, při sečtení všech dílčích pulzů. To znamená, že při dosažení dvaceti pulzů od jednotlivých druhů mincí, pošle mincovník jeden výsledný pulz na jeho výstup.

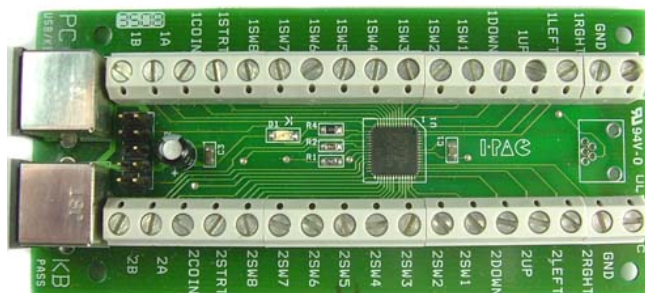
### 3.6 Klávesnicový emulátor

Abych mohl zpracovat výstupní analogové signály z tlačítek a mincovníku, bylo zapotřebí použít zařízení, které převede signály analogové na digitální a pomocí USB rozhraní je zašle do počítače.

První myšlenkou bylo, využít nějaké periferní zařízení, které funguje na podobném principu jako gamepad, klávesnice, joystick atd. Při zamyšlení jsem dospěl k názoru, že musí existovat zařízení sloužící přímo pro tyto účely. Po dlouhém hledání na internetu jsem opravdu jedno takové zařízení našel. Jednalo se o obchod s elektronikou, který sídlí v Brně a tato zařízení vyrábějí pouze na zakázku. Nabízejí pět různých typů emulátorů, které se dělí podle počtu vstupů a samozřejmě cenových nákladů. Jelikož jsem potřeboval minimálně dvacet jedna vstupů (dvacet tlačítek, jeden mincovník) musel jsem tedy objednat bohužel nejdražší model, který údajně zasílají do 14 pracovních dnů. Po měsíci čekání jsem emulátor stále neobdržel, po telefonickém rozhovoru s vedoucím obchodu mi bylo sděleno, že mají technické potíže při výrobě, tudíž jsem byl nucen objednávku zrušit. Po dobrých zkušenostech s mincovníkem a tlačítky z mezinárodních internetových obchodů jsem opět zkusil štěstí zde. Nakonec

jsem byl rád, že nejmenovaná firma v Brně nebyla schopna tento emulátor vyrobit, protože v Anglii se mi ho podařilo zakoupit za poloviční náklady a za přijatelnou dobu doručení.

Emulátor označený jako ARCADE I-PAC2 FS32 disponuje 32 vstupy a vyrábí se jak s USB rozhraním, tak ještě se starším rozhraním PS/2, který se používal k připojení klávesnice a myši k počítači.



*Ilustrace 11: Klávesnicový emulátor ARCADE I-PAC2 FS32*

Každý vstup je opatřen svým vlastním mikroprocesorem, který zajišťuje nulovou odezvu a slouží jako opatření proti kolizím mezi jednotlivými vstupy. Což se stává výhodou u akčních her pro více hráčů na jednom počítači. Emulátor je napájený přímo z USB portu. Pro správnou funkčnost ať už ve Windows, nebo v Linuxu není zapotřebí žádný speciální ovladač, postačí pouze standardní USB ovladač. Zařízení je v podstatě tištěný spoj malých rozměrů, který se díky svorkám na sběrnici pohodlně a jednoduše zapojuje. Výchozí naprogramování emulátoru mi vyhovovalo, nemusel jsem řešit případné přeprogramování. Způsob zapojení tlačítek a mincovníku k zařízení je vyjádřen pomocí následující tabulky.

Tabulka 2: Konfigurační tabulka klávesnicového emulátoru

VSTUPNÍ SVORKA	VÝSTUPNÍ KLÁVESKA	FUNKCE	VSTUPNÍ SVORKA	VÝSTUPNÍ KLÁVESKA	FUNKCE
Hráč 1			Hráč 3		
2SW8	L	doprava	1RGHT	RightArrow	doprava
2SW7	J	doleva	1LEFT	LeftArrow	doleva
2SW6	K	dolů	1UP	UpArrow	dolů
2SW5	I	nahoru	1DOWN	DownArrow	nahoru
1SW8	V	bomba	1SW7	C	bomba
Hráč 2			Hráč 4		
2LEFT	D	doprava	1SW6	X	doprava
2SW1	A	doleva	2DOWN	F	doleva
2SW3	S	dolů	2RGHT	G	dolů
2SW4	W	nahoru	1A	P	nahoru
2SW2	S	bomba	2UP	F	bomba
1SW5	Y	mincovník			

### 3.7 Výběr komponentů a PC sestava

Srdcem celého automatu je počítač, na kterém je spuštěna samotná hra. Vzhledem k zadání a dostupnosti jsem zvolil klasický stolní počítač, který se skládá ze základní desky, procesoru, operační paměti, grafického čipu a pevného disku. Malé nároky na softwarovou náročnost mi umožnily použít starší komponenty.

Tabulka 3: Přehled vybraných komponentů

Základní deska	Asrock ALiveNF6G-VSTA
Procesor	AMD Athlon 64 X2 4000+ EE (socket AM2)
Operační paměť	2x2GB DDR2 (800Mhz) – dual channel
Grafický čip	ATI Radeon HD 3870
Pevný disk	16GB FLASH Kingston DataTraveler 100 G3



Všechny díly sestavy jsem poskládal a uložil do tzv. MiddleTower skříně, která byla opatřena zdrojem o výkonu 350W. Jak je patrné z tabulky, místo obvyklejšího pevného disku, jsem se rozhodl použít flash disk z důvodu úspory elektrické energie a hlučnosti. Abych mohl tuto úpravu uskutečnit, musela vybraná základní deska disponovat USB porty, přes které dochází ke komunikaci mezi flash diskem a samotným počítačem. Dále jsem potřeboval USB port k připojení klávesnicového emulátoru. Posledním požadavkem základní desky byla funkce „Restore on AC / Power Loss“, která zajistí automatické spuštění celého počítače při výpadku elektrického proudu.

### **3.8 Softwarové vybavení**

Abych mohl na počítači spustit svoji videohru, bylo zapotřebí operačního systému, který slouží jako prostředník mezi uživatelem a hardwarem. V dnešní době máme nejrozšířenější dvě platformy. První z nich je pochopitelně Windows od velice známé a proslulé americké společnosti Microsoft. Jako konkurent je už méně rozšířený systém pod názvem Linux. Dále se tu nabízela možnost zvolit úplně netradiční operační systém pro PC, jako je třeba Android, nebo SteamOS. Od samého začátku jsem se přikláněl k operačnímu systému Linux, který je založen na principech unixových systémů. Linux je šířen v tzv. distribucích (Debian, Ubuntu, Fedora, CentOS...), které je snadné nainstalovat. Jedná se o volně šířitelný software, takže je možné ho volně používat, což je největší výhodou tohoto operačního systému. Dalším aspektem mého rozhodnutí byla určitě stabilita, která je podle mých dosavadních zkušeností rozhodně lepší právě na Linuxu.

Zbývalo tedy zvolit pouze vhodnou distribuci a grafické prostředí. Prvním kandidátem se pro mě stal Debian, se kterým jsem ve styku již řadu let a využívám jeho serverové funkce dodnes. Jako grafické prostředí jsem volil Gnome, které je jedním z nejrozšířenějších. Ale dostal jsem se do problému, když jsem chtěl aplikaci spustit. Hra vypsalala na konzolovou řádku obecnou chybovou hlášku, kterou se mi nepodařilo rozluštit. Po poměrně dlouhé době testování různých kombinací distribucí a grafických prostředí mi vyšla vstříc distribuce vycházející právě z Debianu, která se označuje jako Ubuntu 13.10 Saucy Salamander, opět s grafickým rozhraním Gnome 3.12.

V této chvíli jsem měl funkční počítač s nainstalovaným operačním systémem, takže stačilo vytvořenou videohru zkompilem pro linuxové jádro přenést na interní paměť počítače. Jelikož jsou všechny distribuce linuxu nekompromisní na adresářové a souborové oprávnění, musel jsem přidělit příkazem „*chmod a+x nazev\_binarniho\_souboru.x86*“ právo na spouštění binárního souboru hry. Posledním krokem bylo zajistit automatické spuštění aplikace po startu linuxu. To se dalo celkem jednoduše nastavit v grafickém prostředí Gnome, kde se vybere přímo binární soubor, který se má spouštět po startu.

### 3.9 Tvorba aplikace

Centrální logika pro každou hru ve vývojovém nástroji Microsoft XNA tvoří prostředí, kde se hra spustí a dokud nebudou splněny ukončovací podmínky, hra poběží ve smyčce. Každá hra v tomto vývojovém prostředí obsahuje následující základní metody.

- **Game1()** - obecná inicializace (Games.cs)
- **Initialize()** - inicializace hry (Games.cs)
- **LoadContent()** - nahrání (zpravidla grafických) zdrojů (Games.cs)
- **Update()** - čtení uživatelských vstupů, výpočty, testování podmínek pro ukončení hry (Games.cs)
- **Draw()** - kód pro vykreslení grafického výstupu (Games.cs)
- **UnloadContent()** - uvolnění zdrojů (Games.cs)

Pro hru Bomberman jsem navrhl následující třídy:

- **Player**
- **Block**
- **Game**

Třída **Player** obsahuje základní vlastnosti hráčů např. počet smrtí, počet zabití jiného hráče, rychlost pohybu, pozici v herním poli atd. Tato třída má tyto metody:

- `ResetPlayerPos()` - Tato metoda je volaná v konstruktoru této třídy a obsahuje nastavení výchozího umístění hráčů v herním poli.
- `ResetPlayerRound()` - nastavení výchozích hodnot při startu kola
- `ResetPlayerGame()` - vynulování statistik hráčů
- `IsMoveAble()` - kontrola kolizí a pohybu hráče
- `Move()` - Hlavní metoda této třídy, která zajišťuje pohyb hráčů.

Třída **Block** uchovává informace o jednotlivých políčkách. Hra obsahuje tři typy políček.

- **Solid** – Pevný kámen, který se nedá procházet ani uvolnit bombou.
- **Breakable** – Zeď, kterou lze odstranit výbuchem bomb.
- **Air** – Vzduch, kterým může hráč volně projít.

Dále tato třída obsahuje následující metody:

- *`Explode()`* - výbuch políčka
- *`MakeBomb()`* - umístění bomby na příslušné políčko
- *`RunBomb()`* - volání metody `Explode()` - výbuch bomby

Hlavní třídou celé hry je třída **Game**. V inicializační části jsou zde načteny dvourozměrné textury, definovaná velikost herního pole a vytvoření pole pro hráče. Níže je popis jednotlivých metod této třídy.

- *`InitMap()`* - počáteční nastavení herního pole
- *`Update()`* - pravidelně volaná funkce pro aktualizaci průběhu hry
- *`UpdateStartScreen()`* - kontrola vhození mince
- *`UpdateLoginScreen()`* - načtení počtu hráčů

- *UpdateGameScreen()* - Hlavní obrazovka s hrou, která volá metody *UpdateBlock()* a *UpdatePlayers()*.
- *UpdateEndRoundScreen()* - aktualizace statistik po jednotlivých kolech
- *UpdateEndGameScreen()* - aktualizace celkových statistik hry
- *GetWinner()* - Zjišťuje hráče, který vyhrál.
- *UpdateBlock()* - aktualizace políček
- *UpdatePlayers()* - aktualizace hráčů
- *OnGUIStartScreen()* - vykreslení úvodní obrazovky
- *OnGUILoginScreen()* - vykreslení přihlašovací obrazovky
- *OnGUIGameScreen()* - vykreslování hlavní herní obrazovky
- *OnGUIEndRoundScreen()* - vykreslení statistik po jednotlivých kolech
- *OnGUIEndGameScreen()* - vykreslení celkových statistik hry

Při spuštění aplikace hráče uvítá úvodní obrazovka, která se zobrazuje dokud mincovník nepřijme dvacet korun. Následuje přihlašovací obrazovka, kde se navolí počet hráčů. Po přihlášení všech čtyř hráčů se spustí již první kolo samotné hry v herní obrazovce. V případě, že se přihlásí menší počet hráčů, automat vyčká patnáct sekund a hra se zahájí se stávajícím počtem přihlášených hráčů. Hra končí zůstane-li jeden z hráčů naživu nebo naživu nezůstane nikdo. V této situaci se na monitoru zobrazí shrnutí statistik jednotlivých hráčů po odehraném kole a hra pokračuje dalším kolem. Po dosažení tří výher jednoho z hráčů jsou zobrazeny celkové výsledky hráčů a automat se vrací zpět na úvodní obrazovku.

Hra obsahuje konfigurační soubor (viz příloha I), ve kterém jsou definovány hlavní parametry. Díky tomuto souboru se testování, případně úprava různých parametrů, stává velmi pohodlné a není zapotřebí zásahu do zdrojového kódu.

Zdrojové kódy jsou přiloženy na CD v adresáři *source*.

## 4 Závěr

Celý projekt se zpočátku jevil jako celkem snadný, ale ve finále tomu tak nebylo. Nejen, že bylo velice náročné sehnat potřebné a vyhovující součástky, ale také celá konstrukce kabinetu byla velmi náročná.

Stavba celého kabinetu se podařila podle mých představ, ale po několika hodinách testování jsem dospěl k názoru, že by bylo vhodné příště zkonstruovat kabinet vyšší a to z důvodu pohodlnějšího ovládání. Pro lepší manipulaci bych se také zaměřil na menší rozměry celého automatu.

Po odzkoušení tlačítek více hráči jsem dospěl k názoru, že mnou vybrané rozvržení tlačítek se osvědčilo. Při další konstrukci automatu bych zkusil použít místo tlačítek pákové ovladače.

Jelikož každá barva led-diody má odlišnou voltampérovou charakteristiku, je i jiná intenzita podsvícení ovládacích tlačítek pro každého hráče. Z tohoto důvodu bych se snažil u příští stavby lépe promyslet způsob zapojení. Bylo by také vhodné použít nějaký modul, který by zajistil světelné reakce na hru.

Mincovník, který na první pohled připomínal hračku z lega, se v praxi ukázal velmi přesný a spolehlivý. Při testování široké škály mincí se mi ho ani jednou nepodařilo obelstít. Díky jeho ceně a jednoduchosti programování na mě udělal velký dojem.

S klávesnicovým emulátorem jsem neměl nejmenší problém a jsem s ním velice spokojený, ačkoliv byl jednou z nejnákladnějších součástí celého automatu. Stačilo připojit vstupy a následně připojit samotný emulátor k počítači a vše ihned fungovalo. Rozhodně jej budu využívat v budoucnosti v malosériové výrobě.

Největší kámen úrazu nastal ve chvíli kompilace pro linuxové jádro. Jelikož Visual studio od Microsoftu neumožňuje tuto funkci, musel jsem celý kód přepsat do vývojového prostředí, které tuto možnost nabízí. Pro tento účel jsem zvolil MonoDevelop, který implementuje .NET Framework z Windows do Linuxu. Ovšem i zde jsem narazil na problém. Při pokusu o přeložení celé aplikace jsem obdržel

chybovou hlášku, která hlásila problém při implementaci textur. Tento problém se mi odstranit nepodařilo. Rozhodl jsem se zvolit jinou alternativu pro vývojové prostředí. V dnešní době je velice rozšířená platforma Unity3D, která je uzpůsobená pro vývoj her. Unity3D umožňuje přeložit aplikace pro celou řadu nejběžnějších platforem jako je například PC, MAC, Linux, Android, iOS a spousta dalších. Netrvalo dlouho a s tímto příjemným prostředím jsem se seznámil. Kompilace mé hry pro linuxové jádro nyní proběhla v pořádku.

Myslím si, že videohra pro automat byla zvolena správně, nicméně stále je spousta věcí, které lze vylepšit. Přidáním dalších bonusů by se hra stala mnohem hratelnější a zábavnější. Z časové náročnosti celé práce se budu těmito zaležitostmi zabývat v budoucnu, abych docílil co největšího úspěchu z řad uživatelů.

Tato práce mě bavila a přinesla mi mnoho prospěšných poznatků. Celý projekt byl však velice náročný nejen časově, ale i ekonomicky. Stručný přehled finančních nákladů je shrnut v následující tabulce. Doufám, že se tento automat v praxi uplatní a umožní mi malosériovou výrobu.

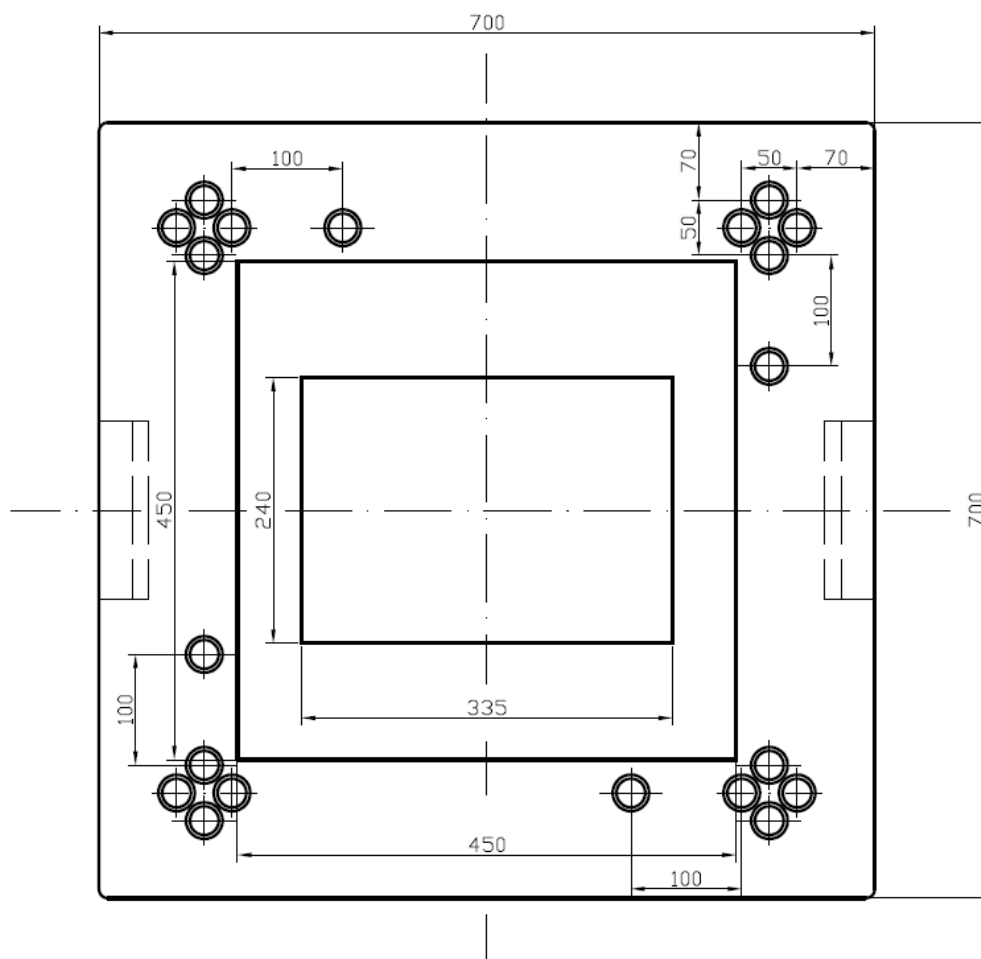
*Tabulka 4: Finanční náklady herního automatu*

<b>Součást kabinetu</b>	<b>Náklady</b>
Stavební materiál	4500 Kč
Madla	200 Kč
Tlačítka	1000 Kč
Mincovník	700 Kč
Klávesnicový emulátor	2100 Kč
Monitor	300 Kč
Flash disk	200 Kč
<b>CELKEM</b>	<b>9000 Kč</b>

## 5 Seznam použité literatury a dalších zdrojů

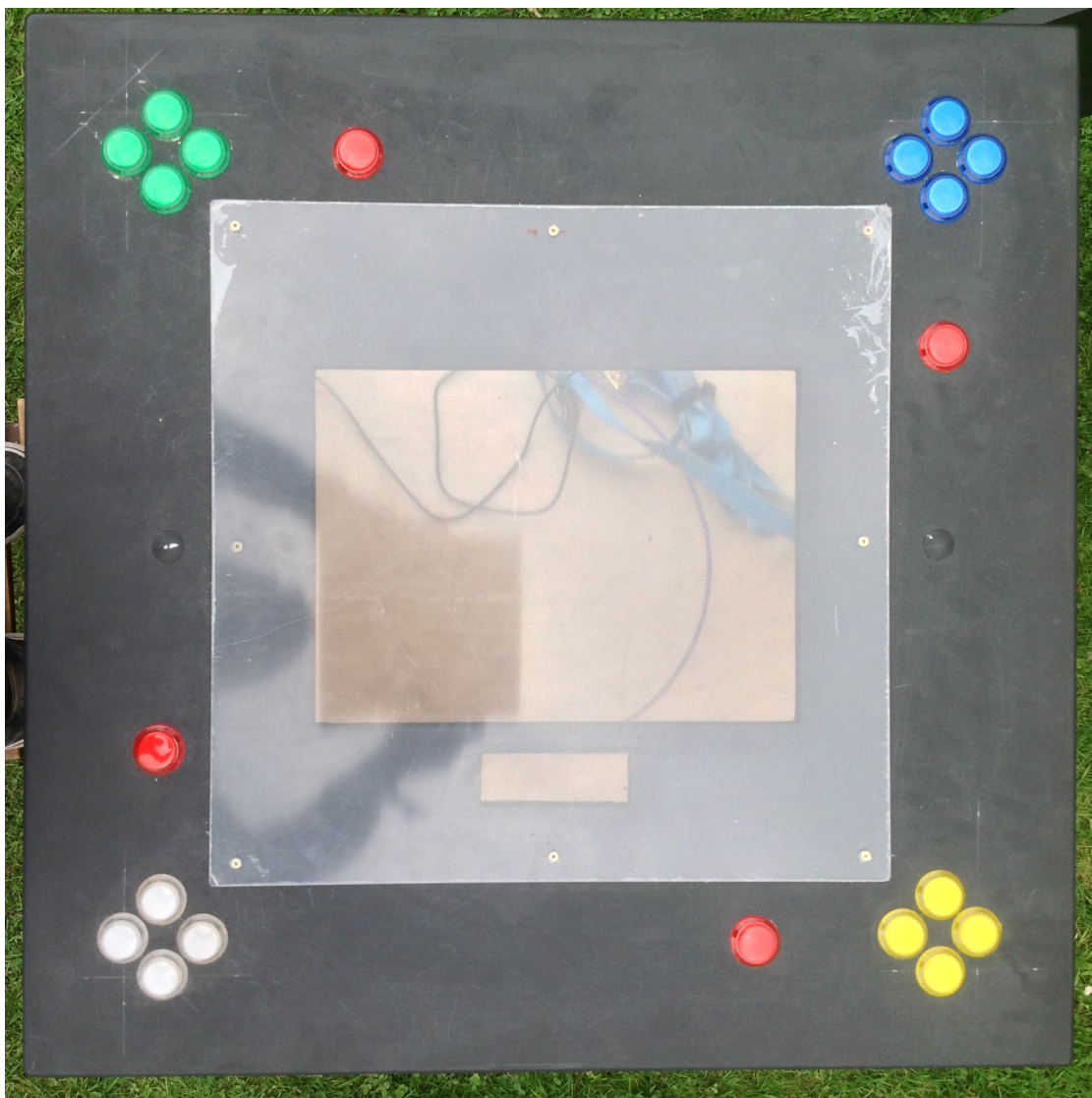
- [1] Hudson. *Arcade Gear - Hudson* [online]. 2012 [cit. 2014-05-12]. Dostupné z: <http://www.arcade-gear.com/Games/Hudson/Hudson.htm>
- [2] Official website. *Liero - Official website* [online]. 2013 [cit. 2014-05-12]. Dostupné z: <http://liero.be/>
- [3] Bulánci. *Bulánci (SleepTeam & Handjoy 2009)* [online]. 2009 [cit. 2014-05-12]. Dostupné z: <http://www.bulanci.cz/>
- [4] VÁVRA, Dan. Arkádové nebe s Danem Vávrou. *Hry - Games.cz* [online]. 2011 [cit. 2014-05-10]. Dostupné z: <http://games.tiscali.cz/tema/arkadove-nebe-s-danem-vavrou-55754>
- [5] Softwarové emulátory - AMIGA REVIEW.online. *AMIGA REVIEW.online* [online]. 1996 [cit. 2014-05-10]. Dostupné z: <http://amigareview.amiga.sk/amiga-review-16/software-emu-16>
- [6] M.A.M.E. *Film - recenze* [online]. 2007 [cit. 2014-05-10]. Dostupné z: <http://www.phweb.wz.cz/mame.html>
- [7] SPIELMANN, Michal a Jiří ŠPAČEK. AutoCAD: názorný průvodce pro verze 2010 a 2011. Vyd. 1. Brno: Computer Press, 2010, 431 s. Názorný průvodce (Computer Press). ISBN 978-80-251-3120-6.

## 6 Přílohy

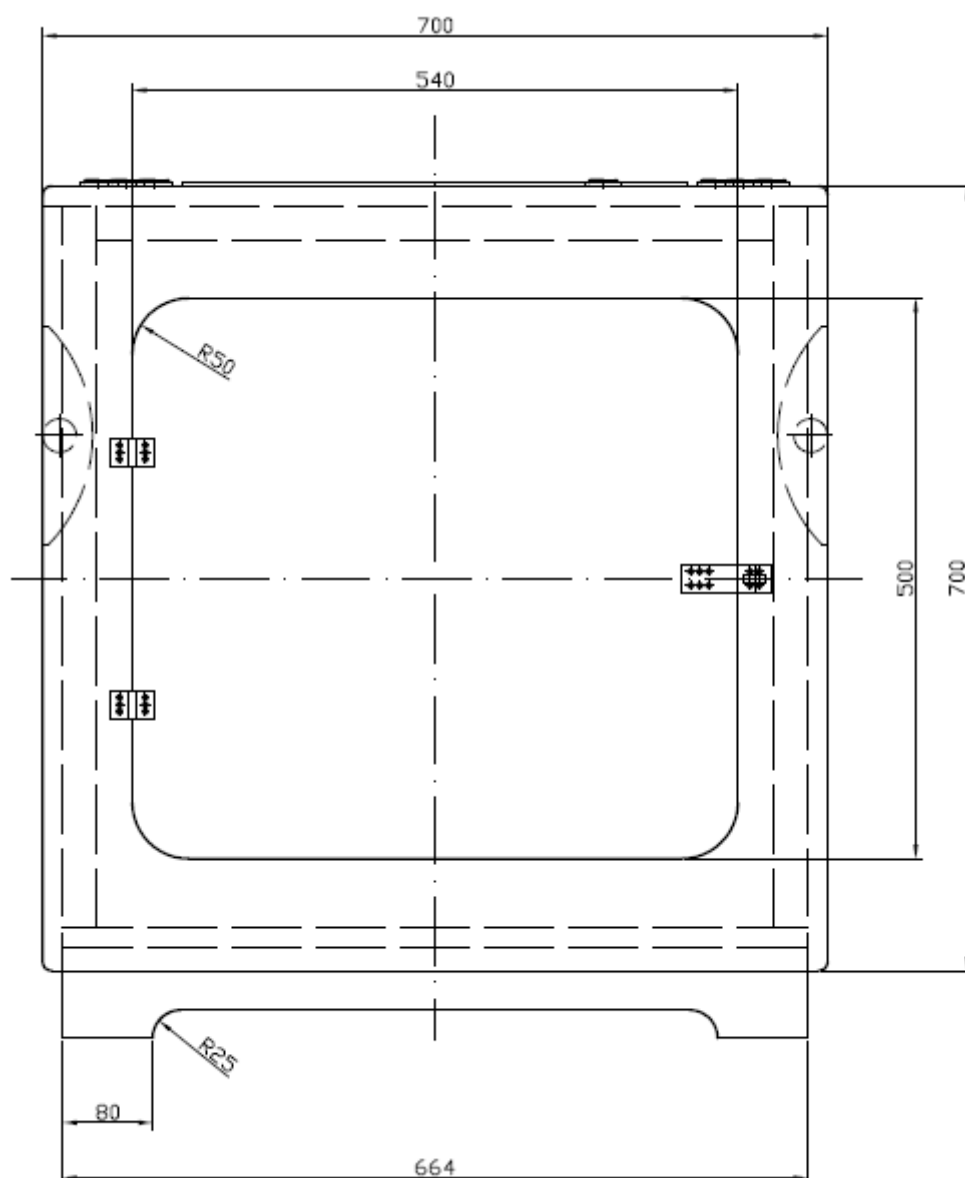


*Příloha A: Technický výkres kabinetu – pohled shora*





*Příloha B: Reálná ilustrace kabinetu - pohled shora*



*Příloha C: Technický výkres kabinetu - pohled zepředu*



*Příloha D: Reálná ilustrace kabinetu - pohled zepředu*





*Příloha F: Reálná ilustrace kabinetu - boční pohled*





*Příloha G: Ukotvení monitoru a počítače*



*Příloha H: Vystužení kabinetu*

```

7   public const int FIELD_SIZE = 64;
8   public const int FIELD_COUNT = 15;
9
10  public const int MAXROUND = 2;
11  public const int MAXWIN = 3;
12
13
14  public const float BOMB_TIME = 3;           //[s]
15  public const float EXPL_TIME = 0.5f;        //[s]
16  public const float LOGIN_TIMEFULL = 60;     //[s]
17  public const float LOGIN_TIMERES = 15;      //[s]
18  public const float LOGIN_TIMEMIN = 1;       //[s]
19  public const float WINWAIT_TIME = 2;        //[s]
20  public const float SHOWDEATHPLAYER_TIME = 2; //[s]
21
22  public const float PLAYER_DEFSPEED = 130;    //[pix/s]
23  public const float PLAYER_MAXSPEED = 250;    //[pix/s]
24  public const float PLAYER_BONUSPEEDSTEP = 30; //[pix/s]
25  public const int PLAYER_DEFBOMB = 1;         //[bomb]
26  public const int PLAYER_MAXBOMB = 4;         //[bomb]
27  public const int PLAYER_DEFEXPLRADIUS = 2;   //[field]
28  public const int PLAYER_MAXEXPLRADIUS = 5;   //[field]
29
30
31  public const float EXPL_RADIUS_BONUS_P = 0.3f;
32  public const float PLAYER_SPEED_BONUS_P = 0.3f;
33  public const float BOMB_BONUS_P = 0.4f;
34
35  //IU
36  public static KeyCode[] KeyRight = new KeyCode[] { KeyCode.K, KeyCode.W, KeyCode.RightArrow, KeyCode.F};
37  public static KeyCode[] KeyDown = new KeyCode[] { KeyCode.J, KeyCode.D, KeyCode.DownArrow, KeyCode.P};
38  public static KeyCode[] KeyLeft = new KeyCode[] { KeyCode.I, KeyCode.S, KeyCode.LeftArrow, KeyCode.X};
39  public static KeyCode[] KeyUp = new KeyCode[] { KeyCode.L, KeyCode.A, KeyCode.UpArrow, KeyCode.G};
40  public static KeyCode[] KeyBomb = new KeyCode[] { KeyCode.V, KeyCode.Q, KeyCode.C, KeyCode.R};
41

```

---

## *Příloha I: Konfigurační soubor hry*